

## Introduction

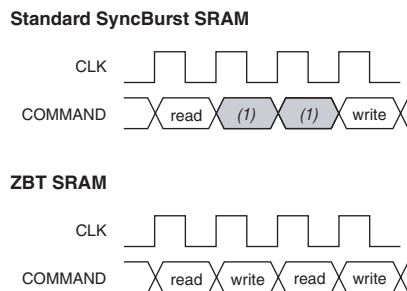
As communication systems require more low-latency, high-bandwidth interfaces for peripheral components, designs need high-throughput memory with efficient bus utilization. The previous generation of static memory types are inefficient when they frequently switch between reading from and writing to the memory. To address this problem, IDT, Micron, and Motorola have developed the new zero-bus turnaround (ZBT) SRAM architecture. To implement this new memory, Altera has developed a ZBT SRAM controller reference design for use with APEX™ II devices.

This application note describes the functionality of the Altera® ZBT SRAM controller reference design and explains the data tree structure, along with installation, compilation, and simulation, of the design file.

## General Description

ZBT SRAM is a synchronous burst (SyncBurst) SRAM with a simplified interface that allows you to take advantage of the full bandwidth by eliminating turnaround cycles and idle cycles between read and write operations. Turnaround cycles, which are necessary for standard SyncBurst SRAM, significantly reduce the available bandwidth. [Figure 1](#) compares the read-to-write transition using SyncBurst SRAM and ZBT SRAM.

**Figure 1. Comparison of Read-to-Write Transitions**



**Note to Figure 1:**

(1) The shaded sections represent no-operation cycles.

ZBT SRAM is available with a pipelined or flow-through interface. In a pipelined interface, the read data is available two clock cycles after the read command is issued, and write data is required two clock cycles after the write command. In a flow-through interface, the data-latency requirement is one clock cycle for read and write operations. Pipelined ZBT SRAM provides faster clock-to-data access, and is therefore clocked faster than flow-through ZBT SRAM. However, flow-through ZBT SRAM provides one clock cycle less latency than the pipelined variety.

Both pipelined and flow-through ZBT SRAM devices can load a new address each clock cycle or advance the device's internal counter. The device can sequence the internal counter in a linear or interleaved fashion. ZBT SRAM supports either LVTTTL or LVCMOS I/O pins.

ZBT SRAM devices have a higher chance of bus contention than standard SyncBurst SRAM devices. Bus contention results when two devices on the bus (for example, the DQ bus) attempt to drive opposite logic values at the same time. During a write operation, the controller drives the data bus and the SRAM receives that data, but during a read operation the controller must tri-state its output drivers and receive the data being driven by the SRAM. If the controller executes a read followed by a write operation and the SRAM does not stop driving the bus before the controller begins to drive the bus, contention occurs if opposite logic levels are being driven. Alternatively, if during a write followed by a read operation the turn-off time of the controller is longer than the turn-on time of the SRAM, contention may result. The effects of short-term contention are minimal; a very small amount of power increase and extra heating occurs. For a more in-depth discussion of bus contention and its effects, refer to the IDT document *Application Brief AN-203 (ZBT SRAMS: System Design Issues and Bus Timing)*, available on the IDT web site at <http://www.idt.com>.

Altera's ZBT SRAM controller provides a simplified interface to ZBT SRAM devices. The controller is available in either Verilog HDL or VHDL, and is optimized for the Altera APEX II device architecture. APEX II devices can interface with ZBT SRAM at 200 MHz.

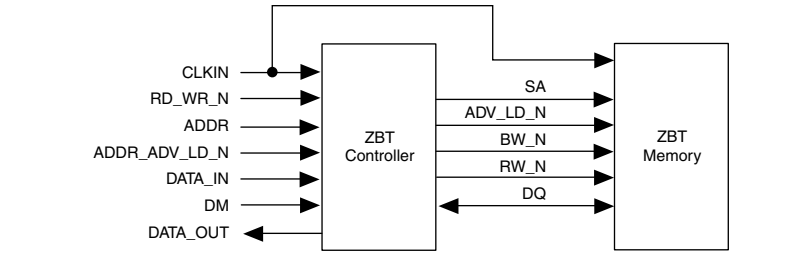
## Reference Design Description

The reference design for the ZBT SRAM memory controller demonstrates generating the ZBT SRAM interface control signals. This design is optimized to run at 200 MHz in APEX II devices. In addition, the APEX II device I/O buffers have a programmable  $t_{ZX}$  delay to avoid bus contention. The  $t_{ZX}$  delay allows a device to quickly release control and slowly take control of a bus. The delay is set by a logic option in the Altera Quartus® II software.

You can implement the Altera ZBT SRAM controller reference design in an APEX II device to provide a simplified interface to a ZBT SRAM device. The reference design implements the ZBT SRAM controller in an EP2A15F672C7 device.

Figure 2 shows a simplified system-level block diagram in which the ZBT SRAM controller connects to a ZBT SRAM module.

**Figure 2. ZBT Controller System-Level Diagram**



### Controller Structure & Operation

The ZBT SRAM controller reference design is made up of five lower-level modules: pipe\_delay, addr\_ctrl\_out, data\_inout, pipe\_stage, and clk\_ctrl. Figure 3 shows a block diagram of the ZBT controller.

**Figure 3. ZBT Controller Block Diagram**

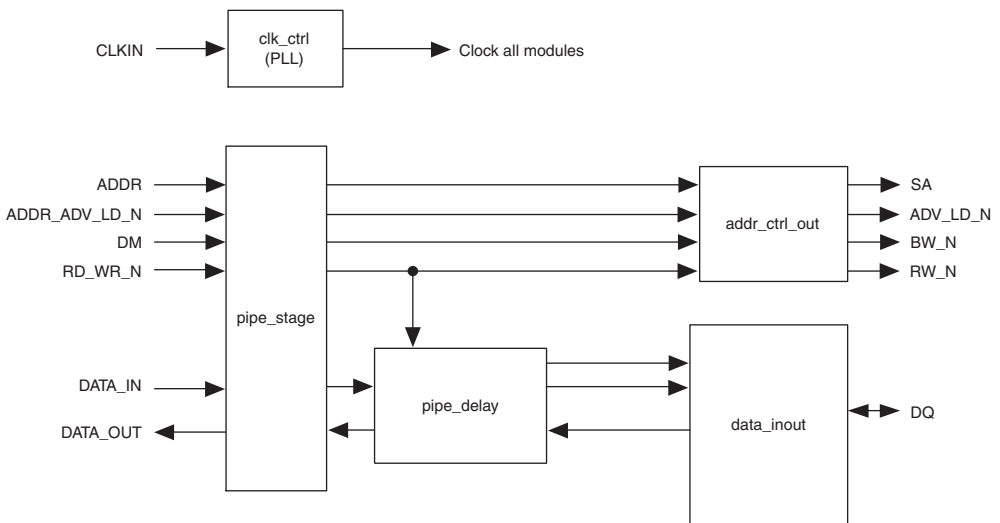


Table 1 describes the function of the five ZBT controller modules.

<b>Module</b>	<b>Function</b>
pipe_delay	Provides the proper delay for the input data, depending on whether the controller is configured for pipelined or flow-through ZBT SRAM
addr_ctrl_out	Registers the address and control signals before outputting them to the ZBT SRAM
data_inout	Registers outgoing data to the ZBT SRAM and provides bidirectional control for the SRAM DQ bus
pipe_stage	Registers the input and output user signals for speed purposes in a stand-alone controller configuration
clk_ctrl	Instantiates a phase-locked loop (PLL) to double the input clock and de-skew the internal and external clocks to the ZBT SRAM

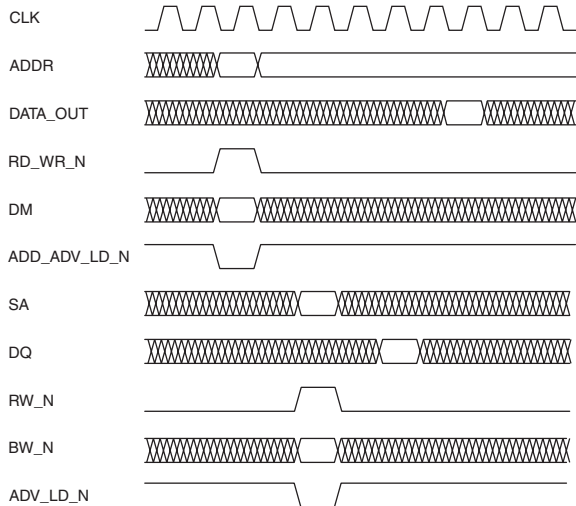
For write operations, the controller aligns the write data to the write command. The controller also handles bus turnaround for transitioning between write and read operations. The following sections provide detailed descriptions of the read and write operations.

## Read Operation

Use the following guidelines for read operations:

- The ADDR, ADDR\_ADV\_LD\_N, and DATAIN signals are set to appropriate values, and RD\_WR\_N is high.
- The data read is available at DATA\_OUT for one clock cycle for both pipelined and flow-through operations.
- The DM bits are not used for read operations.
- Asserting ADDR\_ADV\_LD\_N low causes the ZBT SRAM to write data to the ADDR address. Asserting ADDR\_ADV\_LD\_N high causes the ZBT SRAM internal two-bit counter to increment to the next address (i.e., the internal two-bit counter takes the place of ADDR's lower two bits). The SRAM performs a burst read operation for the number of cycles that ADDR\_ADV\_LD\_N is high.

Figure 4 shows a timing diagram for a single read operation. To perform multiple reads, change the input signals (i.e., ADDR, ADDR\_ADV\_LD\_N, and DM) each cycle while keeping RD\_WR\_N high.

**Figure 4. Read Operation (Pipelined)** *Note (1)***Note to Figure 4:**

(1) Crosshatched areas represent don't care bits.

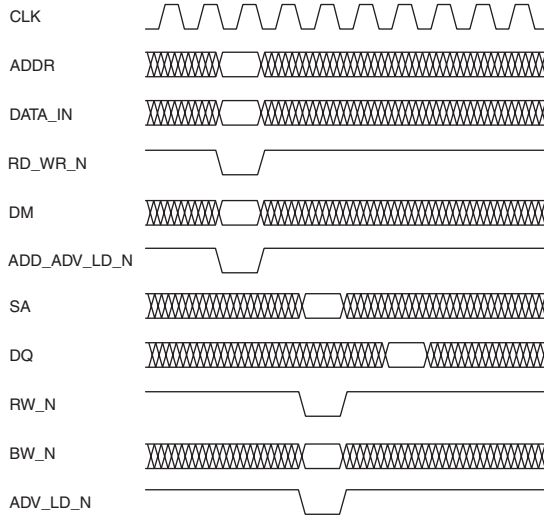
## Write Operation

Use the following guidelines for write operations:

- Set the ADDR, ADDR\_ADV\_LD\_N, DATAIN, and DM signals to the appropriate values, and set RD\_WR\_N low.
- The device clocks data into the ZBT SRAM four clock cycles later for pipelined operation and three clock cycles later for flow-through operation. The controller handles the latency requirements so that data is presented one or two clock cycles after the address and control signals for flow-through or pipelined operations, respectively.
- Set the DM bits to high for each byte line (either eight or nine bits, depending on the SRAM used) written to the SRAM.
- To write data to the ADDR address, designers should assert ADDR\_ADV\_LD\_N low. Asserting ADDR\_ADV\_LD\_N high causes the SRAM internal two-bit counter to increment to the next address performing a burst write operation for the number of cycles that ADDR\_ADV\_LD\_N is high.

Figure 5 shows a timing diagram of a single write operation. To perform multiple writes, change the input signals (i.e., ADDR, ADDR\_ADV\_LD\_N, DATAIN, and DM) each cycle while keeping RD\_WR\_N low.

**Figure 5. Write Operation (Pipelined)** *Note (1)*



**Note to Figure 5:**

- (1) Crosshatched areas represent don't care bits.

## Interface Signals

Table 2 describes the ZBT controller I/O port interface signals.

Signal	Name	Active	I/O Pin	Description
CLKIN	Clock	N/A	Input	System clock
RESET_N	Reset	Low	Input	System reset
ADDR [ASIZE-1..0]	Memory Address	N/A	Input	Memory address for read/write requests; the width is set by ASIZE
RD_WR_N	Read/Write	N/A	Input	ZBT SRAM read/write input; high indicates a read, low indicates a write
ADDR_ADV_LD_N	Address Advance/Load	N/A	Input	Address advance/load input. When high, the ZBT SRAM controller's internal address counter is advanced. When low, a new address is loaded.
DATAIN [DSIZE-1..0]	Input Data	N/A	Input	Input data bus; the width is set by DSIZE
DATAOUT [DSIZE-1..0]	Output Data	N/A	Output	Output data bus; the width is set by DSIZE
DM [BWSIZE-1..0]	Data Mask	High	Input	Masks individual bytes during data write; the width is set by BWSIZE
SA [11..0]	Address Bus	N/A	Output	Address outputs to the ZBT SRAM
ADV_LD_N	Advance/Load	N/A	Output	ADV_LD_N output to ZBT SRAM
BW_N [BWSIZE-1..0]	Byte Write Enables	Low	Output	Controller byte write enables
RW_N	Read/Write	N/A	Output	Controller read/write output; high indicates a read, low indicates a write
DQ [DSIZE-1..0]	Data	N/A	Input/ Output	Bidirectional data port

## Parameters

Table 3 summarizes the parameters of the ZBT SRAM controller function. You can configure these parameters in the file.

Parameter	Description
ASIZE	Specifies the number of address bits in interface.
DSIZE	Specifies the number of data bits in interface.
BWSIZE	Specifies the number of byte enables in interface.
FLOWTHROUGH	Value determines whether the controller operates in flow-through or pipelined mode. For pipelined operation this parameter is set to 0. For flow-through operation, it is set to 1.

## Constraints

To ensure proper logic placement for the controller, use the following constraints in the Quartus II software. These constraints are in the reference file provided by Altera.

- Turn off the **Remove Duplicate Register** logic option (Options & Parameters menu).
- Use the LogicLock™ incremental design capability to place the logic into a block of seven logic array blocks (LABs) within the MegaLAB™ structure at the bottom right corner of the device (MegaLAB block Z4 in the reference design).
- Ensure that the Quartus II software has placed the input, output, and output enable (OE) registers in the I/O element (IOE) for the fastest timing by using the fast I/O register assignments.
- Use the phase shift feature of the APEX II PLL to decrease the  $t_{CO}$  times. However, this feature may increase  $t_{SU}$  times. For example, a phase shift of 4 ns resulted in 3.525-ns  $t_{CO}$  and 1.11-ns  $t_{SU}$  times.

## Clock Generation

The controller clocking scheme maintains consistent and robust high-frequency operation. The Altera ZBT SRAM controller reference design uses one dedicated input clock (`inclk`). This input clock feeds one of the four general-purpose PLLs in the APEX II device. This PLL boosts the input clock and de-skews the internal and external clocks of the ZBT SRAM controller interface.

## Post-Route Performance

The post-route performance results for the ZBT SRAM controller are shown in [Tables 4](#) and [5](#).

**Table 4. Post-Route Performance** *Note (1)*

Device	Internal $f_{MAX}$ (MHz)	$t_{CO}$ (ns)	$t_{SU}$ (ns)	Throughput (Mbytes per second)	Total Logic Elements (LEs)
EP2A15F672C7	200	3.525	1.11	800	67

**Table 5. Data Byte Timing** *Note (1)*

Byte	$t_{CO}$ (ns)	$t_{SU}$ (ns)
DQ [7..0]	3.525	1.11
BW_N [0]	1.58	
DQ [15..8]	3.525	1.11
BW_N [1]	1.58	
DQ [23..16]	3.525	1.11
BW_N [2]	1.58	
DQ [31..24]	3.525	1.11
BW_N [3]	1.58	

*Note to Tables 4 and 5:*

- (1) Timing numbers are preliminary and were calculated using the Quartus II software version 1.1 service pack 1. These timing numbers were calculated using a PLL in the circuit with a phase shift of 4 ns.

## Contention Analysis

The *APEX II Programmable Logic Device Family Data Sheet* states that the PLL clock-to-output buffer disable and enable delays are 4.2 ns and 2.1 ns, respectively. Use the **Increase  $t_{ZX}$  delay to output pin logic** option in the Quartus II software to increase the APEX II device turn-on time by 1.5 ns. IDT's IDT71V3548S133 133-MHz pipelined ZBT SRAM has a 1.5-ns turn-on time and a 1.5 to 3.0-ns turn-off time. Using these values, if the controller performs a read followed by a write operation, there is no contention because the ZBT SRAM turn-off time is faster than the controller turn-on time. For a write followed by a read operation, there is a possibility of no more than 0.6 ns of contention (2.1 ns – 1.5 ns = 0.6 ns).

Likewise, the IDT71V3557S75 100-MHz flow-through ZBT SRAM has a 3.0-ns turn-on time and a 5.0-ns turn-off time. Using these values, if the controller performs a read followed by a write transaction, there will be up to 0.8 ns of contention ( $5.0 \text{ ns} - 4.2 \text{ ns} = 0.8 \text{ ns}$ ). There is no contention for a write followed by a read operation because the turn-off time of the APEX II device is faster than the turn-on time of the ZBT SRAM.

## Getting Started

The Altera ZBT SRAM controller reference design provides solutions for integrating ZBT SRAM into your digital system. This section describes how to install and use the reference design on your PC.

These instructions assume that:

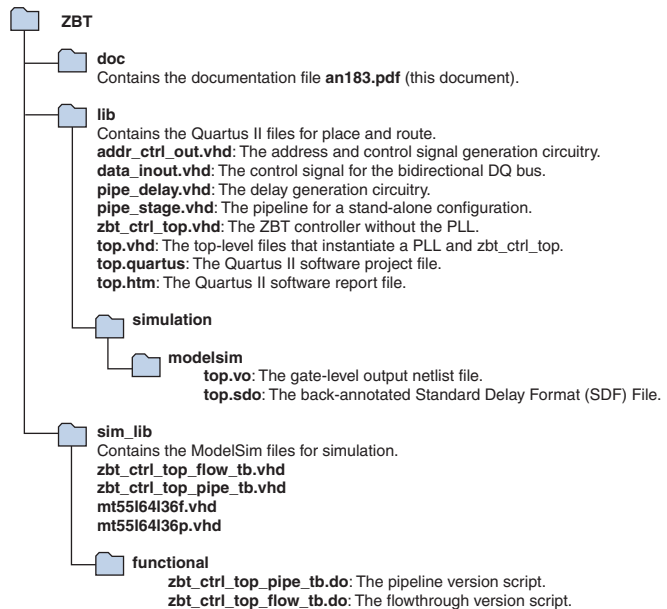
- You are using a PC with the Quartus II software version 1.1 (or higher) installed in the default location.
- You are familiar with the Quartus II software.
- The ModelSim software is installed on your system for simulation.

## Design Installation

Altera provides the ZBT SRAM controller reference design as two compressed files, one for VHDL and one for Verilog HDL. You can download either the Verilog HDL or VHDL version of the reference design from the Altera web site at <http://www.altera.com>. To install the files, perform the following steps:

1. Create a directory on your hard drive to save the files.
2. Save the zip file, **zbt\_rd\_vhdl\_v1.0.0p1.zip**, into the directory you just created. You can delete this file after you finish the installation.
3. Decompress the contents of the **zbt\_rd\_vhdl\_v1.0.0p1.zip** file to the directory created in step 1.

Figure 6 shows the directory structure created for the reference design and displays selected VHDL files (Figure 6 shows VHDL design files only; Verilog HDL files have similar functionality).

**Figure 6. ZBT SRAM Controller Directory File Structure**

## Compile & Simulate Reference Designs

Altera provides the reference design source files to synthesize, place-and-route, and simulate the design. This section takes you through the design flow for the reference design, including compiling and synthesizing in the Quartus II software and simulating in the ModelSim software.

The results for each step are included in the reference design; therefore, you do not need to perform each step unless you have altered the design files. For example, you can view the simulation results without first compiling the design because the Quartus II software place-and-route results are included with the reference design. The compilation and simulation results are in the **lib** and **sim-lib** directories, respectively.

*Compile in the Quartus II Software*

This section describes how to compile and place-and-route the design using the Quartus II software. The **lib** directory contains the Quartus II software version 1.1 project files, including the constraint files needed for the design to meet the required clock frequencies and I/O timing. Altera provides the project files listed in [Table 6](#) in the **lib** directory.

<b>File</b>	<b>Description</b>
<b>pll.vhd</b>	PLL instantiation file created by the Quartus II MegaWizard <sup>®</sup> Plug-In Manager. This file instantiates the parameterized <code>altclocklock</code> function, which generates a PLL in the APEX II device. This PLL is instantiated in the top-level <b>top.vhd</b> file and feeds all the lower-level modules.
<b>addr_ctrl_out.vhd</b>	Registers the address and control bits
<b>data_inout.vhd</b>	Controls the input/output functions of the controller
<b>pipe_delay.vhd</b>	Contains the pipeline delay module for flowthrough and pipelined ZBT SRAM
<b>pipe_stage.vhd</b>	Transmits and receives signals
<b>top.vhd</b>	Instantiates all the files described above and is the top-level file

To compile the Altera-provided project files, use the following steps:

1. Run the Quartus II software.
2. Choose **Open Project** (File menu).
3. Select the **top.quartus** file from the `<work dir>\lib` directory and click **Open**.
4. Choose **Compile Mode** (Processing menu).
5. Choose **Start Compilation** (Processing menu).

### *Simulate in the ModelSim Software*

The `sim_lib` directory contains an HDL testbench file (`zbt_ctrl_top_flow_tb.vhd`) that instantiates the ZBT SRAM controller and the ZBT SRAM model (`<work_dir>\sim_lib\mt55l64l36f.vhd`). The testbench demonstrates the functionality of the controller by first writing into the memory and then reading from the memory and comparing the data. Because the SRAM model is behavioral and does not contain timing information, additional delay is added to the signals to and from the SRAM in the testbench to account for board delay and clock-to-output time. You can model different board-delay scenarios by changing these delays.

Altera provides a script (`zbt_ctrl_top_flow_tb.do`) to perform functional simulation in the ModelSim software. This script creates a work library and pre-compiles the correct simulation libraries for functional simulation. It also compiles the controller source files, the model, and the testbench, and displays the appropriate waveforms. If your files are not in the default installation locations, you must update the script so that the paths point to the locations of the installed reference design and the Quartus II software.

Use the following steps to perform functional simulation:

1. Run the ModelSim version 5.5c software.
2. Change your working directory to the `<work_dir>\sim_lib\functional` directory.
3. Type the following commands in the **Command** window:  
`do zbt_ctrl_top_flow_tb.do`

## Conclusion

ZBT SRAM addresses the need for high-throughput, low-latency static memory and is useful for designs with many successive read and write transactions. The transition latency is minimized because ZBT SRAM does not require turnaround cycles between reads and writes. Altera APEX II devices support ZBT SRAM system frequency up to 200 MHz. The APEX II device I/O buffers are further enhanced by the programmable  $t_{ZX}$  delay logic option to minimize contention during read and write transitions on the device. The Altera ZBT SRAM controller reference design is a ready-to-use memory controller that interfaces APEX II devices with a ZBT SRAM device at up to 200 MHz.



101 Innovation Drive  
San Jose, CA 95134  
(408) 544-7000  
<http://www.altera.com>  
**Applications Hotline:**  
(800) 800-EPLD  
**Literature Services:**  
[lit\\_req@altera.com](mailto:lit_req@altera.com)

Copyright © 2001 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services

